Homework 4

Consider the linear system of equations:

(i) Determine the existence of solutions. (1 point)

(ii) Determine the uniqueness of solutions (if applicable). (1 point)

(iii) Find the set of all solutions. (1 points)

(iv) Find a vector *x* that minimizes the norm of the error

How to Determine if a solution exist ?

Given a system of linear equations represented by the matrix equation $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is the coefficient matrix and $\mathbf{b}$

is the constant vector, we can determine the existence of solutions by comparing the rank of $\mathbf{A}$ and the rank of the augmented matrix[

- *If* $\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{A}|\mathbf{b}]) = \text{n}$ (where n is the number of unknowns), then the system has a unique solution.
- *If* $\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{A}|\mathbf{b}]) < \text{n}$, then the system has infinitely many solutions.
- *If* $\text{rank}(\mathbf{A}) < \text{rank}([\mathbf{A}|\mathbf{b}])$, then the system has no solution.

How to Determine the set of all solutions

No Solution:

- If a system has no solution, it means the equations are inconsistent, and there's no combination of variable values that will satisfy all equations simultaneously.
- In this scenario, there is no set of solutions.

Unique Solution:

- A system has a unique solution when there are no free variables, implying that each variable has a specific value that makes all the equations true.
- This typically occurs when the rank of the coefficient matrix is equal to the number of columns (i.e., the matrix has full rank). In this case, the solution is a single vector in the solution space.

Infinite Solutions:

- When a system has infinitely many solutions, it means that there's a lot of flexibility in how the variables can be chosen.
- The general solution to such a system is a combination of a particular solution and a homogeneous solution. The particular solution, often denoted as $x*$, is one that minimizes the error in a least squares sense.
- The homogeneous solution represents the null space of the coefficient matrix. It includes all vectors that, when multiplied by the matrix, result in the zero vector.

$$\text{Full Set Solution} = \text{Null Space Vectors} + x^*$$

What is the NullSpace?

The null space (or kernel) of a matrix is in linear algebra refers to the set of all vectors that, when multiplied by the matrix, result in the zero vector.

$Ax = 0$

How to determine the size of null space

To determine the number of vectors needed to span the null space (or the dimension of the null space), one can use the relationship:

NullSpace = number of columns in a matrix - rank of a matrix

Problem 1

$$\begin{cases} 4x_2 = 4 \\ 3x_1 + x_2 = -2 \\ 3x_1 - 2x_2 = 3 \end{cases}$$

Hint: For the matrix A in this question, you can compute its pseudo-inverse by hand using the formula $A^+ = (A^T A)^{-1} A^T$

1. Determining solution: Ans: There is no solution
2. There is no uniqueness since there is no solution
3. There is no solution so we dont need to determine the set
4. Finding a vector that minimizes the norm of the vector

Solution to 4

```
% Initialization
clear all;
clc;
close all;

% Define matrix A and vector b
A = [0 4 0; 3 1 0; 3 -2 0];
b = [4; -2; 3];

% Determine rank of matrix A and the augmented matrix
rank_A = rank(A);
augmented_matrix = [A, b];
rank_augmented = rank(augmented_matrix);

% Check the principle to determine the existence of solutions
if rank_A == rank_augmented
    % Compute the least squares solution x* that minimizes the error
    x_star = pinv(A) * b;
    fprintf('x* that minimizes the error is:\n');
    disp(x_star);

    % Determine the nullity (dimension of the null space)
```

```
    nullity_A = size(A, 2) - rank_A;

    % Find a basis for the null space
    N = null(A);

    if nullity_A == 0
        fprintf('The system has a unique solution as the null space is empty.\n');
    else
        fprintf('The system has infinitely many solutions.\n');
        fprintf('Basis for the null space of A:\n');
        disp(N);

        fprintf('\nThe general solution is given by:\n');
        fprintf('x = [\n');
        for i = 1:length(x_star)
            fprintf('%f\n', x_star(i));
        end
        fprintf(']');
        for i = 1:nullity_A
            fprintf(' + c%d*[', i);
            disp(N(:, i));
            fprintf(']');
        end
        fprintf('\nWhere c1, ... are arbitrary scalars.\n');
    end
else
    % Compute the least squares solution x* that minimizes the error
    x_star = pinv(A) * b;
    fprintf('x* that minimizes the error is:\n');
    disp(x_star);
    fprintf('The system has no solution.\n');
end
```

```
x* that minimizes the error is:
    0.2358
    0.4146
         0
The system has no solution.
```

```
% Calculate the error and its norm for the least squares solution
error_star = A * x_star - b;
error_norm = norm(error_star);
fprintf('Norm of the error for x*: %.2f\n', error_norm);
```

```
Norm of the error for x*: 5.00
```

Problem 2

$$\begin{cases} 4x_2 + 4x_3 = 4 \\ 3x_1 + x_2 + 2x_3 = -2 \end{cases}$$

Hint: For the matrix A in this question, you can compute its pseudo-inverse by hand using the formula $A^+ = A^T(AA^T)^{-1}$

1. The system contains a solution and its an infinite amount
2. There is no uniqueness of a solution since theres an infinite amount due to the free variable
3. To find all the sets of solutions we just need to define the free variable
4. Since there is a solution we do not need to compute the error

```
% Define matrix A and vector b
A = [0 4 4;3 1 2];
b = [4 ;-2];

% Determine rank of matrix A and the augmented matrix
rank_A = rank(A);
augmented_matrix = [A, b];
rank_augmented = rank(augmented_matrix);

% Check the principle to determine the existence of solutions
if rank_A == rank_augmented
    % Compute the least squares solution x* that minimizes the error
    x_star = pinv(A) * b;
    fprintf('x* that minimizes the error is:\n');
    disp(x_star);

    % Determine the nullity (dimension of the null space)
    nullity_A = size(A, 2) - rank_A;

    % Find a basis for the null space
    N = null(A);

    if nullity_A == 0
        fprintf('The system has a unique solution as the null space is empty.\n');
    else
        fprintf('The system has infinitely many solutions.\n');
        fprintf('Basis for the null space of A:\n');
        disp(N);

        fprintf('\nThe general solution is given by:\n');
        fprintf('x = [\n');
        for i = 1:length(x_star)
            fprintf('%f\n', x_star(i));
        end
        fprintf(']');
        for i = 1:nullity_A
```

```
            fprintf(' + c%d*[', i);
            disp(N(:, i));
            fprintf(']');
        end
        fprintf('\nWhere c1, ... are arbitrary scalars.\n');
    end
else
    % Compute the least squares solution x* that minimizes the error
    x_star = pinv(A) * b;
    fprintf('x* that minimizes the error is:\n');
    disp(x_star);
    fprintf('The system has no solution.\n');
end
```

```
x* that minimizes the error is:
   -1.1053
    0.6842
    0.3158
The system has infinitely many solutions.
Basis for the null space of A:
   -0.2294
   -0.6882
    0.6882
The general solution is given by:
x = [
-1.105263
0.684211
0.315789
]
 + c1*[
   -0.2294
   -0.6882
    0.6882
]
Where c1, ... are arbitrary scalars.
```

```
% Calculate the error and its norm for the least squares solution
error_star = A * x_star - b;
error_norm = norm(error_star);
fprintf('Norm of the error for x*: %.2f\n', error_norm);
```

```
Norm of the error for x*: 0.00
```